Performance Evaluation of Big Data Processing at the Edge for IoT-Blockchain Applications

Zi Ee Lee
School of Computing Science
University of Glasgow, UK
patrickleeziee@gmail.com

Raphael Liang Hui Chua
School of Computing Science
University of Glasgow, UK
raphaelchualh@gmail.com

Sye Loong Keoh
School of Computing Science
University of Glasgow, UK
SyeLoong.Keoh@glasgow.ac.uk

Yoshihiro Ohba
Toshiba Memory Corporation
Minato-ku, Tokyo, Japan
Yoshihiro.Ohba@toshiba.co.jp

***Abstract*—Internet-of-Things (IoT) utilising sensors is effective in performing continuous monitoring, while Blockchain is ideal in guaranteeing integrity and immutability of these IoT data. There are many challenges in integrating IoT and Blockchain together mainly because IoT devices have limited computational resources, and storage capacity while Blockchain processing incurs high CPU cost and high latency in data transfer. We propose a fully distributed edge computing architecture coupled with an efficient storage system that is based on Non-Volatile Memory express Over Fabrics (NVMeOF) to provide efficient IoT data processing for supply chain management. The data is secured using Blockchain at the edge to ensure traceability, security and non-repudiation in the data. An evaluation of our implementation and performance comparison between NVMeOF and SATA storage interfaces for our IoT-Blockchain architecture is presented.**

I. Introduction

In the age of connected world, Internet-of-Things (IoT) has created a new computing paradigm to automatically sense information without human control in many sectors such as *supply chain management*, environmental monitoring, healthcare, transport and smart manufacturing. By having the capability of continuous monitoring and actuation through the IoT networks, it is now feasible to collect fine-grained data for analysis, detect system anomalies and data tampering in real-time, as well as to predict failures in advance in order to ensure system reliability and safety. However, there are two major concerns that this paper aims to address in this highly connected world paradigm. Firstly, as a result of constant monitoring through IoT sensors, *high volume* of data will be generated and they must be processed in an efficient manner. Traditional approaches to data processing typically involve streaming of data to the cloud services and this has raised concerns such as response time, bandwidth cost, devices' battery life, and data safety and privacy [1]. Secondly, although the data volume is very high, the *veracity* of each piece of the data collected is crucial as its authenticity and trustworthiness will determine the correctness of the data analysis result and affect the predictive analysis adversely.

In this paper, we propose a fully distributed edge computing architecture coupled with an efficient storage system that is based on NVMeOF (Non-Volatile Memory express Over Fabrics) to provide efficient IoT data processing. The data collected through the IoT networks is secured using Blockchain *Hyperledger Sawtooth* at the edge, thus ensuring traceability, security and non-repudiation in the data. Each edge computing node is responsible for processing huge amount of IoT data and acting as highly efficient storage resources [2]. Additionally, each edge node is a blockchain node in a permissioned-private Blockchain network, to ensure strict data protection and integrity between all involved parties. We model our system architecture based on a *food supply chain management* application in which food safety and traceability to the source of contamination are of utmost importance. By deploying IoT technology, the condition and status of food can be closely monitored at all times. Any sign of tampering and change of environmental condition that affected the freshness of the food are logged and subsequently secured using Blockchain at the edge. The proposed architecture is highly efficient as the edge node is equipped with an NVMeOF network storage called Kumoscale [3]. For IoT and Blockchain applications dealing with large-sized data, NVMeOF at the edge significantly outperforms direct attached SATA (Serial Advanced Technology Attachment) with higher throughput for I/O operations and faster blockchain Transaction Execution Rate (TER).

There are two main contributions in this paper. Firstly, we propose a novel edge computing architecture for IoT and blockchain applications for supply chain management. Secondly, this paper provides an in-depth performance comparison between two types of SSD storage interfaces, i.e., NVMeOF and SATA for a specific application, i.e., blockchain for logistic use cases, while in the present works it is known that the maximum interface speed for NVMe and NVMeOF is faster than that for SATA and that NVMe shows better sequential and random access I/O performance than SATA [4] in general (i.e., not specific to any particular application).

This paper is organised as follows: Section II provides a background for Blockchain and related work. Section III presents the IoT-Blockchain architecure for processing big data at the edge network, and Section IV provides details of our implementation. Section V discusses the performance evaluations for different storage systems. We conclude the paper in Section VI with future works.

II. Related Work and Background

This section first presents related deployment of IoT and blockchain technology for supply chain management, followed

by a brief background information on blockchain and smart contract.

A. Related Work

Liu *et al.* [5] proposed a blockchain-based hybrid cloud and edge architecture for vehicular networks. Each vehicle is an intermediary between the cloud and the sensor. It is responsible for processing huge amount of sensor data and performs analysis during collaborative operations. The use of edge computing provides higher accuracy on measurement and reduction of latency as compared to the traditional cloud.

Xiong *et al.* [6], proposed a concept of mobile Blockchain where mobile devices act as Blockchain nodes which offload mining computation tasks to edge computing servers, and provided performance evaluation in terms of mining resource management, whereas our work is more focused on the supply chain management use cases.

SmartEdge [7] is an architecture to support effective offloading for IoT devices. When implementing Blockchain application, the heavy Blockchain processing tasks are devolved to a compute node, while a Raspberry Pi is used as the data node. It has been shown that IoT applications with blockchain capability are now feasible with edge computing.

B. Blockchain and Smart Contract

A Blockchain is a database or a distributed ledger that is shared across a network of computers. When a transaction occurs between two parties, it is added into the latest block which will be validated through the Blockchain's consensus. The validated block will be hashed and appended to the previous block. Once a transaction is added to the chain, it becomes immutable to changes.

Smart contracts are self-executing contracts that are written in Solidity and stored in Blockchain. It contains all the rules, conditions, expiry dates and all other relevant information needed to govern the interaction between all involved parties. The code is automatically executed once the terms have been fulfilled. The smart contract is stored in the Blockchain and then triggered by sending a transaction to the contract's address which will then be validated onto a block and broadcast throughout the network [8]. The respective nodes will then execute the code on their Ethereum Virtual Machine (EVM) [9] to validate the transaction sent by the miner.

III. Architectural Overview

This section presents the architecture of big data processing at the edge using *food supply chain management* as a use case.

A. Supply Chain Logistics Application

Figure 1 illustrates the proposed architecture for big data processing at the edge for IoT and Blockchain applications – *a food supply chain logistic application*. Each entity in the supply chain, i.e., *farm production, Warehouse, Distributor, Retail* and *Restaurant* is an edge computing node, which also acts as a Blockchain node. Only the entities involved are part of this private Blockchain network, and this means that only

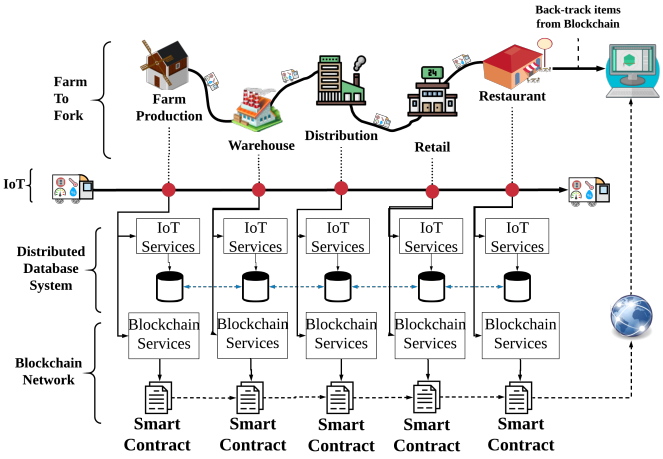

Fig. 1: Overview of Supply Chain Logistics Application

these entities are authorised to view and validate Blockchain transactions via its dedicated *Blockchain Service*. Due to the sensitivity of the information logged, we argue that by having each edge node running a *Blockchain Service*, it ensures that the data remain confidential between the involved parties, while offering transparency and traceability.

An *IoT Monitoring Service* is used to continuously monitor the food condition during transit from one point to another in the supply chain. The environmental context information is logged continuously by the sensors, and any sign of unusual events such as tampering and contamination of food are captured through video recording during the transportation process. When the delivery truck arrives at each checkpoint in the supply chain, it offloads the monitored data and video recording onto the edge computing node via its *IoT Service*. The sensor data collected are then written onto the Blockchain's distributed ledger, while the video content is written onto a *Distributed Database System* (DDBS) and then replicated across all edge nodes. All these processing must be done swiftly in an efficient and effective manner.

Each node's DDBS and storage services are mounted on NVMeOF network storage system to provide fast, low-latency and efficient processing of data. NVMeOF is designed to have only less than 10μs additional latency to locally attached NVMe for which average latency is approximately 300μs [4]. With this, we advocate that it is more efficient to store the large video content directly onto DDBS, and then writing the hash of the content onto the Blockchain to ensure its integrity. This improves the performance of the system significantly.

The *Blockchain Service* includes the use of smart contracts for monitoring of food delivery. A smart contract is formed between the *Retailer* and the *Distributor* prior to the start of delivery process. It is then deployed onto the Blockchain network to store status information of the delivery from the IoT trucks. The smart contract will have all product information encoded, such as *order id, batch id, product name, product quantity* and *payment info*.

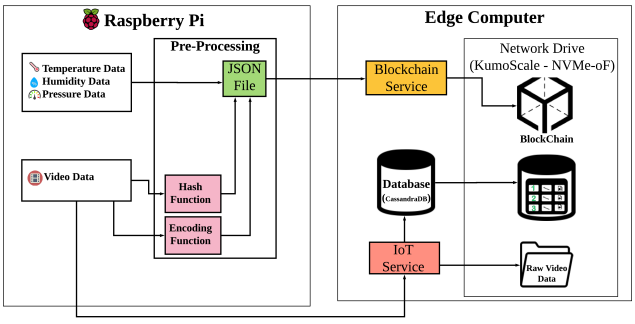

Fig. 2: Flow of data transfer from RP3 to Edge Node

B. IoT Monitoring System

The *IoT monitoring service* is responsible for logging the environmental data in the delivery truck, this includes *temperature, humidity, pressure, location, time, signs of tampering* and *video recording* of the delivery process. A Raspberry Pi Model 3 (RP3) with several sensors is used to perform the monitoring in the delivery truck.

Assuming that the food is to be delivered from the *Distributor* to the *Retailer*. When the truck leaves the *Distributor*, the RP3 will detect the disconnection of WiFi network from the *Distributor* edge, and subsequently activates all sensors to start logging the sensor data. Similarly, the camera will start the video recording while the truck is in transit to the *distributor* site. When the delivery truck arrives at the *Retailer* site, the RP3 will automatically connect to the *Retailer*'s edge network. While unloading the food or goods, all the sensor data and the video recording of the delivery process are also transferred to the edge for further processing. Figure 2 shows the interaction between the *IoT Monitoring System* with the edge computing node that is connected to NVMeOF network storage system.

C. Pre-Processing of IoT Data

The sensor data and the video recording collected by the *IoT Monitoring Service* essentially serve as fine-grained traces of the changing condition of the monitored item (i.e., food) over time throughout the delivery process. It is important that these data are secured, and they can be used as evidence of tampering and to identify the source of food contamination, thus allowing the involved parties to trace back to the time at which the food was tampered with as well as the location that the tampering took place. Prior to sending these data to the edge computing node, to be written into the Blockchain via its *Blockchain Service*, some pre-processing are performed, namely:

1) The video recording is hashed using SHA-256 to generate a unique hash value to be included together with the sensor data to be written into the Blockchain. This is because the size of video recording is typically very large and it is not logical to write the entire video content in the Blockchain. However, in order to ensure the integrity of the recorded video, the hash value of the video content is written onto the Blockchain instead.
2) The file identifier of the video recording is encoded as well, thus providing a pseudo-file ID to be written onto the Blockchain.
3) The hash value of the video and its encoded file name are appended to the sensor data file. They are then sent to the *Blockchain Service* to be written as blockchain transactions. Each dataset <*temperature, humidity, pressure, date, time, location, status of tampering*> is written as a transaction in the Blockchain. As there is only one video recording from one point to another, the hash value of the video together with its encoded pseudo-file ID is written to the Blockchain as a single transaction after all the sensor data have been fully committed.

D. IoT Service

As it is inefficient to store the entire video content in the Blockchain, our architecture proposes that the raw video content is first stored in an NVMeOF network drive mounted on the edge computing node, and then replicated to all other edge computing nodes involved in the system using a *Distributed Database System* (DDBS), i.e., CassandraDB. By using NVMeOF, each edge node has potentially *unlimited* storage space, and the storage can be easily expanded to allow for scalable deployment of the proposed solution. With this, when inspecting the monitored data, the corresponding video content can first be retrieved from the NVMeOF network storage connected to the edge node itself if available. Otherwise, the video content is obtained through the DDBS. The integrity of the video can then be verified by checking the hash value logged in the Blockchain.

The *IoT Service* performs the following when it receives the video content from the *IoT Monitoring Service*:

1) Establishes a connection with DDBS (e.g., CassandraDB), and split the video content into multiple tiny blobs of 1MB each for fast insertion. Each blob is indexed and inserted into DDBS. It is automatically replicated across all edge nodes in the food supply chain management.
2) The entire video content is also stored in NVMeOF storage at the edge node. This would speed up the retrieval of the un-splited video locally.

E. Blockchain Service

The deployed smart contracts between the *Retailer* and the *Distributor* existed throughout the food supply chain. Each of the entities has the exact same copy of the smart contract. To meet the strict requirements of the food supply chain, product information has to be stored in the smart contracts. In the event of food contamination, tracing back to the source would be easier due to the use of smart contract for back-tracking from the top of the supply chain.

Once the IoT truck reaches the supply chain location, the product information (e.g., food) is sent to the edge node and this triggers the initialization of the *Blockchain Service* by executing the smart contract. The smart contract stores the information as transactions collected from the *IoT Monitoring*

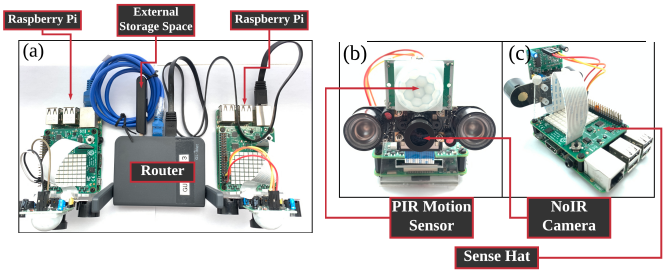

Fig. 3: Hardware components of *IoT Monitoring Service*

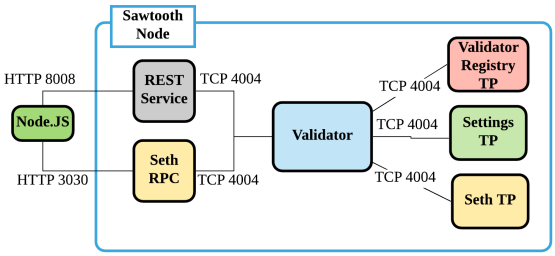

Fig. 4: Hyperledger Sawtooth Seth Implementation

Service. Each dataset <*temperature, humidity, pressure, date, time, location, status of tampering*> is written as a transaction. It is then validated by the blockchain's consensus and thus ensuring the integrity of the dataset.

As the proposed system is based on permissioned-private Blockchain, the validation of blocks is based on Proof-of-Elapsed-Time (PoET) consensus algorithm. The use of PoET ensures that each node in the supply chain will have the chance to validate the block of transactions. The PoET consensus mechanism will assign each node a random countdown timer assuming that each of them has a trusted core based on Intel's Software Guard Extensions (SGX). The first node with its timer expired will be the Blockchain node to be responsible for validating the next block [10]. In this case, only one validator is chosen based on PoET to validate the signed transaction.

IV. Implementation

A. IoT Monitoring Device - Raspberry Pi 3

Figure 3 shows the integrated hardware components used in the *IoT Monitoring Service*. Part (a) shows the top view of two RP3 connected to a shared volume through a wireless router that can be deployed inside the delivery truck. Part (b) shows the front view of the RP3 where the Passive Infrared (PIR) Motion Sensor was used to detect human motion, if detected this implies a sign of food tampering, as well as an No Infrared (NoIR) camera to perform video recording in the truck compartment during the delivery process. A recording of these signs of food tampering can be used as security evidence in case that there was a dispute raised by the receiving entity. Lastly, Part (c) shows the SenseHat integrated with an RP3 to collect environmental data in the truck compartment, ensuring that the temperature and humidity of the food remained in healthy level. This ensures the freshness of the food during the delivery process.

B. Blockchain Hyperledger Sawtooth with Seth

Seth was recommended through the use of Docker for development as it is easier to deploy in each of the mounted volumes at the edge computing node. The Sawtooth Blockchain network consists of a *Seth client, validator, Seth* and *PoET transaction processor, Seth RPC* and *REST API*. The Sawtooth comprises two type of nodes, namely genesis node and normal node. There is only one genesis node, while the other normal nodes need to be connected to the genesis node.

In Figure 4, NodeJS was used to invoke the API calls from the *REST API Service* and *Seth RPC*. The REST API service then retrieves the transaction information from the *validator* while *Seth RPC* focuses on the interaction with the smart contract. The validator is supported by three transaction processor (TP), *settings-tp*, *Seth-tp* and *validator registry-tp*. The *settings-tp* stores all the validators configuration. The *Seth-tp* handles the execution and creation of smart contracts. The *validator registry-tp* is used by PoET consensus to verify when a validator tries to submit a block to the Blockchain.

In order to interact with Seth, external account on the network must be created. Creating an account is equivalent to generating a new private key for Seth. Seth accepts *secp256k1* private key generated with OpenSSL library with or without encryption. Each account has a nonce value which is initialised to either zero or one. Once the account has been created, an account address is returned by Seth.

The validator and PoET consensus are the core of the Blockchain applications. For PoET configurations, the initial wait time was set to 100 seconds and the target wait time was set to 25 seconds. The validator's max transaction per block was set to 30.

C. NVMeOF and KumoScale

NVMeOF is a network protocol which enables a set of NVMe initiators and a NVMe target residing in different machines to be connected over a high-speed network protocol stack such as Remote Direct Memory Access (RMDA) over Ethernet. The NVMe target machine has directly attached SSDs. KumoScale [3] is a shared accelerated storage software for NVMeOF. It seamlessly integrates with multiple orchestration and provisioning frameworks to enable dynamic management of compute and storage allocation matched to each application instance with high performance and low latency. Kumoscale supports multiple storage volumes (a. k. a. multi-tenant), where each storage volume can be configurable either to be shared among a set of NVMe initiators or to be exclusively allocated to a specific NVMe initiator.

In this paper, we used KumoScale NVMeOF with RDMA over 100Gb Ethernet as the network protocol stack for implementation and performance evaluation, where three NVMeOF initiator machines acting as edge nodes and one NVMeOF target machine were connected through a 100Gb Ethernet switch. In a real system, NVMeOF initiator machines that are

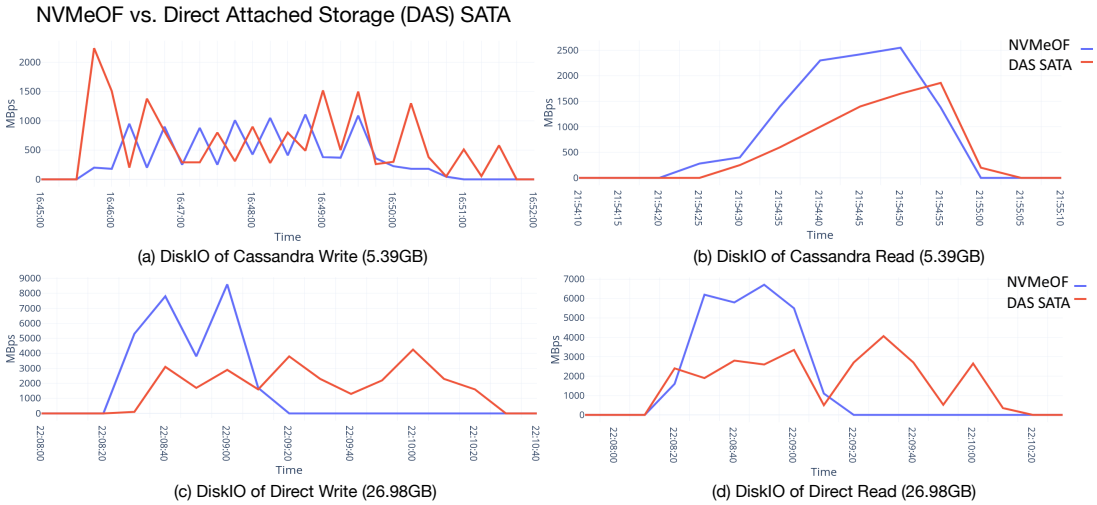

Fig. 5: I/O Performance of Direct Read/Write and Cassandra on NVMeOF and SATA

geographically located in a longer distance than the maximum transmission distance of the underlying communication media are expected to be connected to different NVMeOF target machines.

V. Performance Evaluation

We provide performance comparison on the IoT and Blockchain applications at the edge (i) to compare I/O performance between two different SSD storage interface types when processing high volume of sensor data and Blockchain transactions, and (ii) to provide a preliminary CPU performance when executing Blockchain transactions using the proposed edge computing architecture. The two different SSD interfaces are NVMeOF and directly attached SATA .

A. I/O

Figure 5(a) and (b) show the file I/O performance for accessing the video content stored in DDBS, i.e., CassandraDB in multiple blobs (1MB each).

As shown in Figure 5(a) and Table I, the max writing speed for SATA was higher than NVMeOF, however SATA had a lower max reading speed as compared to NVMeOF. The mean write speed between SATA and NVMeOF shows that the performance of NVMeOF (328 MBps) was comparable to SATA (452 MBps) and NVMeOF had a significantly better read mean speed as compared to SATA (600 MBps vs. 207 MBps). It is observed that there was no significant performance difference in write and read speeds between NVMeOF and SATA. This might be due to CassandraDB performing data replication across all nodes in its network cluster as each tiny blob of 1MB was written into every node in order to facilitate fast replication. A 20GB video file for example, would be split into multiple blobs to be stored in CassandraDB, and hence this had incurred significant amount of I/O operations. Note that both the read/write speed were rather low, i.e., < 3GBps when using CassandraDB on both NVMeOF and SATA.

TABLE I: Performance Comparison of CassandraDB Write/Read – 5.39GB

Storage	Max Write	Max Read	Write Mean	Read Mean
NVMeOF	1.109 GBps	2.587 GBps	328 MBps	600 MBps
SATA	2.242 GBps	1.861 GBps	452 MBps	207 MBps

We also observed that SATA had additional file access operations for accessing Docker container cache which is always stored in SATA SSD. This would actually increase SATA's I/O speed. By subtracting the I/O speed for accessing Docker container cache from the SATA read and write speeds, the I/O performance for accessing CassandraDB via SATA and NVMeOF was nearly the same.

TABLE II: Performance Comparison of Direct Write/Read – 26.98GB

Storage	Max Write	Max Read	Write Mean	Read Mean
NVMeOF	8.596 GBps	6.715 GBps	2.080 GBps	2.076 GBps
SATA	4.250 GBps	4.062 GBps	1.500 GBps	1.499 GBps

As shown in Figure 5(c) and (d), and Table II, for direct reading/writing of large sized file of 26.98GB, NVMeOF had a much higher maximum and mean read/write speeds than SATA. NVMeOF's maximum write speed was almost double than that for SATA and its maximum read speed was approximately 3GBps more than SATA. Even though NVMeOF is a network-storage, it is observed that its I/O performance was significantly better than SATA for IoT data with a large data size.

B. CPU Usage of Blockchain Processing

Blockchain applications incur intensive CPU usage as they need to continuously validate the blocks and then replicate the blocks across all nodes in the Blockchain network. We deployed the IoT-Blockchain architecture on an edge node with the following CPU specification: Intel(R) Xeon(R) Gold 5115 CPU @ 2.40GHz. As shown in Figure 6(a), processing

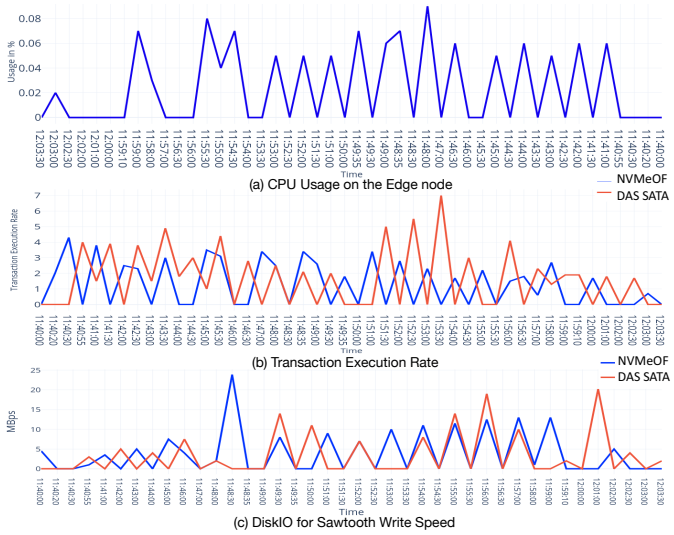

Fig. 6: *Blockchain Service* Performance on NVMeOF & SATA

blockchain transactions on a dedicated edge node had very low CPU usage with only 9% utilisation. If our architecture were to be deployed in the cloud, its performance will be significantly degraded as the cloud CPU is typically shared by multiple virtual machine instances known as resource pooling. We therefore argue that IoT-blockchain applications should be deployed on the edge with dedicated CPU and storage resources.

C. Blockchain Transaction Execution Rate and I/O

Figure 6(b) and Table III show the transaction execution rate (TER) of blockchain applications on NVMeOF and SATA. TER is defined as the number of Blockchain transactions executed in one second. Although NVMeOF had a lower maximum TER, its average TER was higher than SATA. Blockchain has a queue mechanism to prevent overloading. In each second, the blockchain will write the transactions to the block which will be validated in the Blockchain. Once the block has been filled with transactions, the Blockchain will proceed to store the transactions in a new block.

TABLE III: Transaction Execution Rate of Sawtooth

Types of storage	Average	Maximum
SATA	1	7
NVMeOF	1.26	4.30

With regards to I/O cost incurred by Blockchain processing, there is no significant difference between the read/write speed on NVMeOF or SATA. The average write speed for both storage systems was approximately 2MBps as shown in Figure 6(c), with a maximum write speed of $\approx$ 20MBps. Similarly, as the size of the block is fixed and small, writing and validating the blocks did not have much impact on the write speed.

VI. Conclusions and Future Work

This paper has presented a new edge computing architecture utilising NVMeOF (i.e., Kumoscale) as its storage system to support efficient big data processing for IoT-Blockchain applications. Our research has revealed that with an efficient storage system such as NVMeOF and dedicated CPU at the edge, huge volume of data can be transferred from IoT network to the edge, and subsequently processed and protected using blockchain in a highly efficient manner. Even though NVMeOF was used as a networked storage at the edge, its high I/O performance has resulted in faster processing, hence lower latency in the *food supply chain management application*. It is noteworthy that NVMeOF significantly outperformed direct-attached storage SATA for writing large volume of IoT data, while maintaining an on-par performance for Blockchain processing with SATA.

Our preliminary results have shown that edge computing has an advantage for big data processing for IoT-Blockchain applications over cloud-based architecture. It is important for a future work to conduct a more in-depth comparison and analysis between the two architectures to analyze the performance gains of the edge in terms of latency, I/O, CPU usage and Blockchain processing cost.

References

[1] J. Cao, Q. Zhang, and W. Shi, *Edge Computing: A Primer*. Springer, 2018.

[2] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the internet of things: A case study," *IEEE Internet of Things Journal*, vol. 5, no. 2, 2018.

[3] "KumoScale". (2018) Available: https://kumoscale.toshiba-memory.com/. [Accessed: 2019-04-07].

[4] "NVMeOF". (2014) Available: https://nvmexpress.org/wp-content/uploads/NVMe-Webcast-Slides-20141111-Final.pdf. [Accessed: 2019-04-07].

[5] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, May 2018.

[6] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When Mobile Blockchain Meets Edge Computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.

[7] K.-L. Wright, M. Martinez, U. Chadha, and B. Krishnamachari, "SmartEdge: A Smart Contract for Edge Computing," in *1st Inter. Workshop on Blockchain for the IoTs, Halifax, Canada*, August 2018.

[8] B. K. Mohanta, S. S. Panda, and D. Jena, "An overview of smart contract and use cases in blockchain technology," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, July 2018, pp. 1–4.

[9] R. Norvill, B. B. F. Pontiveros, R. State, and A. Cullen, "Visual emulation for ethereum's virtual machine," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2018, pp. 1–4.

[10] "Hyperledger-Sawtooth". (2019) Available: https://sawtooth.hyperledger.org/docs/core/nightly/0-8/introduction.html. [Accessed: 2019-04-04].